

ST0254 – Organización de computadores Práctica 1: Métodos HASH – Bucket Sort - Entrega final

Alexander Ortiz Uribe (aortizu@eafit.edu.co)
Sebastian Garces Montoya (sgarces5@eafit.edu.co)
Juan David Henao Zapata (jhenaoz@eafit.edu.co)

15 de febrero de 2014

Resumen

El presente documento contiene de forma corta y simple, el enunciado de la primera práctica que realizarán los estudiantes del curso ST0254 – Organización de computadores en el semestre 2014-1.

1. PALABRAS CLAVE

Assembler, HASH, Simuproc, Algoritmo Bucket Sort

2. Assembler, HASH, Simuproc, Algoritmo Bucket Sort

2.1. Ordenamiento por casilleros

El ordenamiento por casilleros (bucket sort o bin sort, en inglés) es un algoritmo de ordenamiento que distribuye todos los elementos a ordenar entre un número finito de casilleros. Cada casillero sólo puede contener los elementos que cumplan unas determinadas condiciones. En el ejemplo esas condiciones son intervalos de números. Las condiciones deben ser excluyentes entre sí, para evitar que un elemento pueda ser clasificado en dos casilleros distintos. Después cada uno de esos casilleros se ordena individualmente con otro algoritmo de ordenación (que podría ser distinto según el casillero), o se aplica recursivamente este algoritmo para obtener casilleros con menos elementos. Se trata de una generalización del algoritmo Pigeonhole sort. Cuando los elementos a ordenar están uniformemente distribuidos la complejidad computacional de este algoritmo es de $O(n)$.

El algoritmo contiene los siguientes pasos:

1. Crear una colección de casilleros vacíos

2. Colocar cada elemento a ordenar en un único casillero
3. Ordenar individualmente cada casillero
4. Devolver los elementos de cada casillero concatenados por orden

3. OBJETIVOS

3.1. Objetivo General

Realizar el proceso de construcción de un programa que lea desde una posición de memoria dada, una lista de números y los ordene mediante dos tablas hash.

3.2. Objetivos Específicos

- a Crear un programa que lea desde una posición de memoria x , una lista de números con las siguientes características
 - 1 Si en la posición x hay un 0, significa que el número a leer es un entero y se debe ordenar en el hash de los enteros
 - 2 Si el número en la posición x es un 1, significa que el número a leer es un flotante y se debe ordenar en el has de los flotantes
- b Mostrar de manera independiente, los números enteros y flotantes ordenados de menor a mayor.

4. Problemas encontrados

- Conocimiento nulo en el desarrollo con el lenguaje de programación Assembler.

- Poco conocimiento en el desarrollo con la herramienta de trabajo “Simuproc”, además cuando se ejecuta un código con loop infinito se llena la memoria ram del ordenador obligando al cierre forzado del programa y pérdida de información.
- Problemas para la comparación de flotantes y el método hash para flotantes.
- Problemas con el manejo de números negativos.
- Instrucciones limitadas de Assembler en el simulador
- Recorrer arreglos

5. Solución a problemas encontrados

Al comenzar con el desarrollo y entendimiento de la práctica, comenzamos con una aproximación “top down” seleccionando el algoritmo de ordenamiento y comenzando su implementación en pseudocódigo, al realizar el cambio al lenguaje ensamblador encontramos grandes dificultades y fue requerida la ayuda del monitor y la documentación en línea de “simuproc”.

Al encontrarnos con los flotantes nos dimos cuenta que no se podía usar las mismas operaciones que con los enteros, además eran requeridos dos registros para poder realizar las operaciones requeridas con estos números, fue necesario la consulta con el monitor, además de la documentación del programa, el cual cuenta con algunos problemas en operaciones con números de punto flotante.

Al ejecutar un programa con errores y sin una instrucción de fin clara o inalcanzable este es cerrado a la fuerza por el sistema operativo, ocasionando una pérdida de información que en muchos casos retraso en gran manera el desarrollo de la práctica, para solucionar el inconveniente se acordó usar un sistema manejador de versiones (Github) además de varios respaldos y guardar el trabajo realizado antes de ponerlo en ejecución.

El recorrido de los arreglos fue uno de los problemas iniciales de la práctica, el cual fue el más fácil de solucionar, por medio de un correo al docente, monitor y al desarrollador del simulador, buscando la solución para el recorrido de los arreglos.

6. Funcionamiento del programa

Cuando el programa inicia, comienza con el recorrido de datos los cuales se separan en enteros y punto flotante, posteriormente se hace el ordenamiento de casilleros (método hash) y luego se procede a organizar la información de los casilleros usando el algoritmo de ordenamiento Bubble Sort

7. Posibles mejoras

Como posibles mejoras que se le pueden hacer al programa:

- Verificar el funcionamiento con decimales y buscar el desarrollo de un solo algoritmo de ordenamiento para enteros y flotantes.
- Usar una versión del lenguaje ensamblador para un manejo más sencillo de arreglos y variables